

DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

```
CCCCCCCCC 000000 MM MM MM MM AAAAAA NN NN DDDDDDDD
CCCCCCCCC 000000 MM MM MM MM AAAAAA NN NN DDDDDDDD
CC          00          MMMM MMMM MMMM MMMM AA AA NN NN DD DD
CC          00          MMMM MMMM MMMM MMMM AA AA NN NN DD DD
CC          00          MM MM MM MM MM MM AA AA NNNN NN DD DD
CC          00          MM MM MM MM MM MM AA AA NNNN NN DD DD
CC          00          MM MM MM MM MM MM AA AA NN NN DD DD
CC          00          MM MM MM MM MM MM AA AA NN NN DD DD
CC          00          MM MM MM MM MM MM AA AA NN NN DD DD
CC          00          MM MM MM MM MM MM AA AA NN NN DD DD
CC          00          MM MM MM MM MM MM AA AA NN NN DD DD
CCCCCCCCC 000000 MM MM MM MM AA AA NN NN DDDDDDDD
CCCCCCCCC 000000 MM MM MM MM AA AA NN NN DDDDDDDD
```

```
....
....
....
....
```

```
LL          IIIIII SSSSSSSS
LL          IIIIII SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLLL IIIIII SSSSSSSS
```

COMMAND
Table of contents

- PROCESS NEXT COMMAND

D 7

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00

Page 0

(3)	141	PROCESS NEXT COMMAND
(4)	170	PROCESS REST OF COMMAND
(6)	653	CALL LOGINOUT TO ABORT THE PROCESS
(7)	683	EOD/DECK COMMANDS
(8)	703	CHECK FOR CONTROL Y/C AST PENDING
(9)	724	ENABLE/DISABLE CONTROL Y/C AST'S
(10)	758	FLUSH COMMAND BUFFER
(11)	788	PROCESS FOREIGN COMMAND
(12)	831	GET INTERNAL ROUTINE INDEX

```
0000 1 .TITLE COMMAND - PROCESS NEXT COMMAND
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28 PROCESS NEXT COMMAND
0000 29
0000 30 D. N. CUTLER 22-MAR-77
0000 31
0000 32 MODIFIED BY:
0000 33
0000 34 V03-018 CWH3018 CW Hobbs 28-Jul-1984
0000 35 Fix HWS0093 to avoid loop on a control/y.
0000 36
0000 37 V03-017 HWS0093 Harold Schultz 23-Jul-1984
0000 38 If interactive and at indirect level zero, unconditionally
0000 39 clear the execute-only procedure flag.
0000 40
0000 41 V03-016 HWS0061 Harold Schultz 19-Apr-1984
0000 42 Restore locked keypad state only after PRC_V_CNTRL
0000 43 is cleared.
0000 44
0000 45 V03-015 HWS0020 Harold Schultz 06-Mar-1984
0000 46 When parsing a command which begins with a verb, ignore
0000 47 leading blanks when expanding a symbol definition.
0000 48
0000 49 V03-014 HWS0008 Harold Schultz 13-Feb-1984
0000 50 Remove obsolete code for setting up prompt block in PRC.
0000 51
0000 52 V03-013 PCG0015 Peter George 06-Feb-1984
0000 53 Fix EOJ so that it works even if LOGOUT is deleted.
0000 54 Fix bug in force exiting a privileged image.
0000 55
0000 56 V03-012 PCG0014 Peter George 12-Jan-1984
0000 57 Fix typo.
```


0000	58	:	
0000	59	:	
0000	60	:	V03-011 PCG0013 Peter George 03-Jan-1984
0000	61	:	Set FLUSH bit in flush routine.
0000	62	:	
0000	63	:	V03-010 PCG0012 Peter George 16-Aug-1983
0000	64	:	Disable indirection when looking ahead for null nodes.
0000	65	:	
0000	66	:	V03-009 PCG0011 Peter George 20-Jul-1983
0000	67	:	Support null node names.
0000	68	:	
0000	69	:	V03-008 PCG0010 Peter George 15-Jun-1983
0000	70	:	Fix broken branch.
0000	71	:	Remove extraneous labels.
0000	72	:	Remove references to PRC_L_TAB_VEC.
0000	73	:	
0000	74	:	V03-007 PCG0009 Peter George 27-May-1983
0000	75	:	Have command dispatch table branch to JMP's.
0000	76	:	
0000	77	:	V03-006 PCG0008 Peter George 30-Apr-1983
0000	78	:	Move token sorting routine to DESCRVAL.
0000	79	:	Clear WRK_B_PARMSUM.
0000	80	:	
0000	81	:	V03-005 PCG0007 Peter George 01-Apr-1983
0000	82	:	Move GOTO scanning to GOTO module.
0000	83	:	
0000	84	:	V03-004 PCG0006 Peter George 15-Feb-1983
0000	85	:	Update to new structure level.
0000	86	:	Remove SETBIT WRK_V_NOSTAT in EOD.
0000	87	:	Clear PTR_L_ENTITY and PTR_B_NUMBER when doing DCL\$GENDESCR.
0000	88	:	Sort new larger PTR blocks.
0000	89	:	Add DCL\$LOOKUP_INTERNAL.
0000	90	:	Init recall and keypad fields in WRK block.
0000	91	:	
0000	92	:	V03-003 PCG0005 Peter George 16-Nov-1982
0000	93	:	Use WRK_C_SCRSTKSIZ and WRK prompt string descriptor.
0000	94	:	
0000	95	:	V03-002 PCG0004 Peter George 28-Oct-1982
0000	96	:	Use PRC data structure for prompt string.
0000	97	:	
0000	98	:	V03-002 PCG0003 Peter George 19-Oct-1982
0000	99	:	Fix batch logout accvio caused by DCL\$ABORT.
0000	100	:	Use DCL\$ABORT for DCL\$EOJ.
0000	101	:	
0000	102	:	V03-001 PCG0002 Peter George 30-Sep-1982
0000	103	:	Move common command parsing code to DCL\$PARSE_COMMAND
0000	104	:	in PARSENT. Initialize WRK_L_PROMPTRTN, WRK_B_VALLEV,
0000	105	:	and WRK_B_CMDOPT.
0000	105	:	--

```

0000 107 :
0000 108 : MACRO LIBRARY CALLS
0000 109 :
0000 110 :
0000 111 $SCLITABDEF ; DEFINE TABLE STRUCTURES
0000 112 PTRDEF ; DEFINE RESULT PARSE DESCRIPTOR FORMAT
0000 113 SYMDEF ; DEFINE SYMBOL ENTRY OFFSETS
0000 114 PRCDEF ; DEFINE PROCESS WORK AREA
0000 115 WRKDEF ; DEFINE COMMAND WORK AREA
0000 116 $CLMSGDEF ; DEFINE ERROR/STATUS VALUES
0000 117 $DEVDEF ; DEFINE DEVICE CHARACTERISTIC BITS
0000 118 $PSLDEF ; DEFINE PROCESSOR STATUS FIELDS
0000 119 $RABDEF ; DEFINE RAB OFFSETS
0000 120
00000000 121 .PSECT DCL$ZDATA,BYTE,RD,NOWRT
0000 122
54 55 4F 47 4F 4C 00' 0000 123 LOGOUT: .ASCII 'LOGOUT'
06 0000
0000 124
00000000 0007 125 ROUTINES_LEN = 0
0007 126
0007 127 .MACRO INTIMAGE NAME
0007 128 $NAM1$ = ^A/%EXTRACT(0,4,NAME)/!^X80
0007 129 $NAM2$ = ^A/%EXTRACT(4,4,NAME)/
0007 130 .LONG $NAM1$
0007 131 .LONG $NAM2$
0007 132 ROUTINES_LEN = ROUTINES_LEN + 8
0007 133 .ENDM
0007 134
0007 135 ROUTINES:
0007 136 INTIMAGES ; GENERATE TABLE OF ROUTINE NAMES
01B7 137
00000000 138 .PSECT DCL$ZCODE,BYTE,RD,NOWRT
0000 139

```

```
0000 141 .SBTTL PROCESS NEXT COMMAND
0000 142 :+
0000 143 : DCL$RESTART - PROCESS NEXT COMMAND
0000 144 :
0000 145 : THIS ROUTINE IS THE CENTRAL COMMAND PARSE AND DISPATCH ROUTINE FOR THE
0000 146 : STARLET DCL$ COMMAND LANGUAGE INTERPRETER. IT INITIALIZES FOR THE NEXT
0000 147 : COMMAND, READS AND PARSES THE COMMAND FOR SYNTACTIC CORRECTNESS, AND THEN
0000 148 : EXECUTES THE APPROPRIATE IMAGE.
0000 149 :
0000 150 : INPUTS:
0000 151 :
0000 152 : FP = ADDRESS OF WRK STRUCTURE
0000 153 : R11 = ADDRESS OF PRC STRUCTURE
0000 154 :
0000 155 : OUTPUTS:
0000 156 :
0000 157 : COMMAND IS READ, PARSED, AND AN IMAGE IS ACTIVATED.
0000 158 :-
0000 159 : .ENABL LSB
0000 160 :
0000 161 DCL$RESTART::
5E F486 CD 9E 0000 162 MOVAB WRK_K_LENGTH(FP),SP ;RESTART ENTRY POINT
0005 163 ;ALLOCATE COMMAND IMPURE AREA
0005 164 ; AND RESET STACK POINTER
5A 5D D0 0005 164 MOVL FP,R10 ;SET BASE ADDRESS OF WRK
FFFF'CF 9E 0008 165 MOVAB W^DCL$HYPHEN-1,- ;SET TO FORCE INPUT ON NEXT GET
F48E CA 000C 166 WRK_L_CHARPTR(R10)
0B 11 000F 167 BRB 1$ ;PROCESS COMMAND
0011 168
```



```
0011 170 .SBTTL PROCESS REST OF COMMAND
0011 171 :+
0011 172 : DCL$CMDSTART - PROCESS REST OF COMMAND
0011 173 :
0011 174 : THIS ROUTINE IS THE CENTRAL COMMAND PARSE AND DISPATCH ROUTINE FOR THE
0011 175 : STARLET DCL$ COMMAND LANGUAGE INTERPRETER. IT INITIALIZES FOR THE COMPLETION
0011 176 : OF THE CURRENT COMMAND LINE, READS AND PARSES THE COMMAND FOR SYNTACTIC
0011 177 : CORRECTNESS, AND THEN EXECUTES THE APPROPRIATE IMAGE.
0011 178 :
0011 179 : INPUTS:
0011 180 :
0011 181 : FP = ADDRESS OF WRK STRUCTURE
0011 182 : R11 = ADDRESS OF PRC STRUCTURE
0011 183 : WRK_L_CHARPTR(FP) = POINTER INTO COMMAND INPUT BUFFER
0011 184 : WRK_G_BUFFER(FP) = COMMAND EXPANSION BUFFER
0011 185 : WRK_G_INPBUF(FP) = COMMAND INPUT BUFFER
0011 186 :
0011 187 : OUTPUTS:
0011 188 :
0011 189 : COMMAND IS READ, PARSED, AND AN IMAGE IS ACTIVATED.
0011 190 :-
0011 191 :
0011 192 DCL$CMDSTART:: :COMMAND PROCESSING ENTRY POINT
0011 193 :
0011 194 : ALLOCATE AND INITIALIZE COMMAND SCRATCH WORK AREA
0011 195 :
0011 196 : MOVAB WRK_K_LENGTH(FP),SP :ALLOCATE COMMAND IMPURE AREA
0016 197 : : AND RESET STACK POINTER
0016 198 : MOVL FP,R10 :SET BASE ADDRESS OF WRK
0019 199 1$: MOVZBW PRC_B_PROMPTLEN(R11),- :SET PROMPT LENGTH
001D 200 : WRK_W_PMPLEN(R10)
0020 201 : MOVAB PRC_W_PMPCTRL(R11),- :SET PROMPT ADDRESS
0024 202 : WRK_L_PMPADDR(R10)
0027 203 : CLRB PRC_B_CONTINUE(R11) :SET REGULAR PROMPT
002B 204 : TSTL PRC_L_INDEPTH(R11) :ARE WE AT COMMAND LEVEL 0?
002E 205 : BNEQ 2$ :NO, CHECK CTRL-Y FLAGS
0030 206 :
0030 207 : BBS #PRC_V_MODE,PRC_W_FLAGS(R11),5$ :SKIP IF BATCH MODE
0035 208 : CLRB PRC_B_EXONLYL(R11) :CLEAR EX-ONLY FLAG
0039 209 : BRB 5$ :USE '$ ' PROMPT
003B 210 2$: BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),5$ :USE '$ ' IF CONTROL/Y LEVEL
0040 211 : BBS #PRC_V_CNTRL,PRC_W_FLAGS(R11),5$ : (ALSO IF CTRL/Y JUST HIT)
0045 212 : MOVB #^A/^/,PRC_B_CONTINUE(R11) :NO -- PROMPT STRING = '$ '
004B 213 5$: MOVAB WRK_G_BUFFER(R10),WRK_L_EXPANDPTR(R10) :SET ADDRESS OF EXPANSION BUF
0052 214 : MOVAB WRK_G_RESULT(R10),WRK_L_RSLNXT(R10) :SET ADDRESS OF RESULT PARSE TAB
0058 215 : MOVAB DCL$INPUT,WRK_L_READRTN(R10) :SET RECORD INPUT ROUTINE
0061 216 : MOVAB DCL$INPUT,WRK_L_PROMPTRTN(R10) :SET PROMPT ROUTINE
006A 217 : MOVAB DCL$SPECIAL,WRK_L_SPECRTN(R10) :SET SPECIAL PROCESSING ROUTINE
0073 218 : MOVAB DCL$CHARERROR,WRK_L_ERRORRTN(R10) :SET ERROR HANDLER ROUTINE
007C 219 : MOVAB DCL$ERRORMSG,WRK_L_SIGNALRTN(R10) :SET ERROR SIGNALING ROUTINE
0084 220 : CLRW WRK_W_FLAGS(R10) :RESET COMMAND FLAGS
0087 221 : CLRB WRK_B_VALLEV(R10) :RESET VALUE LEVEL
008A 222 : CLRB WRK_B_CMDOPT(R10) :RESET OPTION NUMBER
008D 223 : CLRL WRK_L_RSLEND(R10) :ZERO ADDR OF LAST TOKEN DESC
0090 224 : CLRL WRK_L_QUABLK(R10) :ZERO QUABLK
0093 225 : CLRL WRK_L_PROPTR(R10) :ZERO PROPTR
0096 226 : CLRL WRK_L_IMAGE(R10) :ZERO ADDRESS OF IMAGE NAME
```

SE F486 CD 9E 0011 196
5A 5D D0 0016 197
00F0 CB 9B 0016 198
F99E CA 0019 199
00F1 CB 9E 0020 201
F9A2 CA 0024 202
00F3 CB 94 0027 203
5C AB D5 002B 204
OB 12 002E 205
16 68 AB 06 E0 0030 207
012D CB 94 0035 208
10 11 0039 209
OB 68 AB 0B E0 003B 210
06 68 AB 01 E0 0040 211
00F3 CB 5F 8F 90 0045 212
F486 CA F492 CA 9E 004B 213
BA AA F9B6 CA 9E 0052 214
F9AA CA 00000000'EF 9E 0058 215
F9A6 CA 00000000'EF 9E 0061 216
F9B2 CA 00000000'EF 9E 006A 217
F9AE CA 00000000'EF 9E 0073 218
D6 AA 00000000'EF 9E 007C 219
F0 AA B4 0084 220
C4 AA 94 0087 221
C3 AA 94 008A 222
B6 AA D4 008D 223
CA AA D4 0090 224
C6 AA D4 0093 225
E2 AA D4 0096 226


```

      D2 AA D4 0099 227      CLRL WRK_L_PAROUT(R10)      ;ZERO PAROUT
      D1 AA 94 009C 228      CLRB WRK_B_MINPARG(R10)     ;ZERO MINPARG
      D0 AA 94 009F 229      CLRB WRK_B_MAXPARG(R10)     ;ZERO MAXPARG
      CE AA 94 00A2 230      CLRB WRK_B_PARMCNT(R10)     ;ZERO PARMCNT
      CF AA 94 00A5 231      CLRB WRK_B_PARMSUM(R10)     ;ZERO PARMSUM
    68 AB 24 AA 00A8 232      BICW #PRC_M_DISABLE!-      ;RE-ENABLE CONTROL Y/C'S
      00AF CB 44 8F 8A 00AC 233      PRC_M_IND,PRC_W_FLAGS(R11) ;RE-ENABLE @ PROCESSING
      00AF CB 44 8F 8A 00AC 234      BICB #PRC_M_RUNDEF!-  ;CLEAR IMAGE RUN DEFAULT FLAG
      00B2 235      PRC_M_FLUSH,PRC_B_FLAGS2(R11) ;RESET FLUSH IN PROGRESS FLAG
      00B2 236
      00B2 237
      00B2 238
      00B2 239
      C5 AA 94 00B2 240      CLRB WRK_B_RECALLCNT(R10)    ;ZERO RECALL COMMAND COUNT
    012F CB D0 00B5 241      MOVL PRC_L_RECALLPTR(R11),-    ;COPY RECALL PTR
      EA AA 00B9 242      WRK_L_RECALLPTR(R10)
      00BB 243
      00BB 244
      00BB 245
      00BB 246
      00BB 247
      00BB 248
      00BB 249
      00BB 250
      00BB 251
      00BB 252
      20 68 AB 01 E5 00BB 253      BBCC #PRC_V_CNTRL,PRC_W_FLAGS(R11),11$ ;BRANCH IF NOT CONTROL Y/C
      FF3D' 30 00C0 254      BSBW DCL$LOCKED STATE ;RESTORE LOCKED KEYPAD STATE
      F896 CA 94 00C3 255      CLRB WRK_G_INPBUF(R10) ;ABORT TWO-PART COMMAND IN PROGRESS
    51 00B8 CB D0 00C7 256      MOVL PRC_L_ONCTLY(R11),R1 ;GET ADDRESS OF ACTION COMMAND
      0B 13 00CC 257      BEQL 10$ ;BR IF NO ACTION OUTSTANDING
      F896 CA 50 81 9A 00CE 258      MOVZBL (R1)+,R0 ;GET THE ADDRESS AND COUNT
      61 50 28 00D1 259      MOVC R0,(R1),WRK_G_INPBUF(R10) ;MOVE COMMAND INTO BUFFER
      2F 11 00D7 260      BRB 15$ ;CLEAR GOTO, THEN PROCESS COMMAND
      37 11 00D9 261 10$: SETBIT PRC_V_YLEVEL,PRC_W_FLAGS(R11) ;SET CONTROL Y/C LEVEL
      00DE 262      BRB 20$
      00E0 263
      00E0 264
      00E0 265
      00E0 266
      00E0 267
      56 FF1D' 30 00E0 268 11$: BSBW DCL$LOCKED STATE ;RESTORE LOCKED KEYPAD STATE
      00E0 CB 7E 00E3 269      MOVAQ PRC_Q_COMMAND(R11),R6 ;GET DESCRIPTOR OF COMMAND LINE
    10 00AF CB 00 E4 00E8 270      BBSC #PRC_V_CMD,PRC_B_FLAGS2(R11),13$ ;BRANCH IF USER COMMAND SETUP
    23 00AF CB 01 E5 00EE 271      BBCC #PRC_V_CHAIN,PRC_B_FLAGS2(R11),20$ ;BRANCH IF NOT USER CHAIN
      56 00D8 CB 7E 00F4 272      MOVAQ PRC_Q_IMAGENAME(R11),R6 ;GET DESCRIPTOR OF IMAGE TO INVOKE
      00F9 273      SETBIT PRC_V_RUNDEF,PRC_B_FLAGS2(R11) ;SET RUN DEFAULT
      F896 CA 04 B6 66 28 00FE 274 13$: MOVC (R6),R4(R6),WRK_G_INPBUF(R10) ;SET STRING IN BUFFER
      FEF8' 30 0105 275      BSBW DCL$ALLDEACMD ;GO DEALLOCATE THE SPACE
      0108 276
      0108 277
      0108 278
      0108 279
      0108 280
      F895 CA 9E 0108 281 15$: MOVAB WRK_G_INPBUF-1(R10),- ;SET POINTER BEFORE RECORD
      F48E CA 010C 282      WRK_L_CHARPTR(R10)
    03 68 AB 04 E5 010F 283      BBCC #PRC_V_GOTO,PRC_W_FLAGS(R11),20$ ;CLEAR GOTO FLAG
```

```
FEE9' 30 0114 284 BSBW DCL$DEALGOTO ;DEALLOCATE GOTO SYMBOL
      0117 285
      0117 286
      0117 287 : IF A HANGUP PENDING, THEN ABORT THE PROCESS.
      0117 288
03 68 AB 0C E1 0117 289 20$: BBC #PRC_V_HANGUP,PRC_W_FLAGS(R11),22$ ;IF SET, HANGUP PENDING
      03E5 31 011C 290 BRW DCL$ABORT ; LOG THE PROCESS OUT
      011F 291
      011F 292
      011F 293 : IF A CHAINED IMAGE WAS SPECIFIED, THEN PROCESS THE IMAGE IMMEDIATELY AND
      011F 294 : AS A FOREIGN COMMAND.
      011F 295
06 00AF CB 02 E1 011F 296 22$: BBC #PRC_V_RUNDEF,PRC_B_FLAGS2(R11),30$ ;BRANCH IF NOT CHAIN IMAGE
      FEDB' 30 0125 297 BSBW DCL$MARK ;MARK CURRENT PARSE POSITION
      012A 31 0128 298 BRW FORVERB ;PARSE AS FOREIGN COMMAND
      012B 299
      012B 300
      012B 301 : BEGIN PARSING INPUT
      012B 302
      012B 303 : IF WE DON'T HAVE A COMMAND LINE YET, THEN FORCE AN INPUT TO HAPPEN.
      012B 304 : LOOP TO RESTART IF THE INPUT IS NULL. STRIP OFF THE LEADING '$'
      012B 305 : IF IT IS PRESENT.
      012B 306
      FED2' 30 012B 307 30$: BSBW DCL$SETNBLK ;POINT TO NEXT NONBLANK CHARACTER
      4E 13 012E 308 BEQL 50$ ;IF EQL NULL LINE
50 24 91 0130 309 35$: CMPB #^A/$/,RO ;DOLLAR SIGN?
      18 12 0133 310 BNEQ 40$ ;IF NEQ NO-PROCESS THE COMMAND
F486 CA F492 CA 9E 0135 311 MOVAB WRK_G_BUFFER(R10),WRK_L_EXPANDPTR(R10) ;RESET EXPANSION BUFFER POINT
      FEC1' 30 013C 312 BSBW DCL$MOVCHAR ;PUT DOLLAR SIGN IN BUFFER
      7C AB DD 013F 313 PUSHL PRC_L_INDCLOCK(R11) ;SAVE CURRENT INDIRECT CHANGE COUNT
      FEBB' 30 0142 314 BSBW DCL$SETNBLK ;POINT AT NEXT NON-BLANK CHARACTER
      37 13 0145 315 BEQL 50$ ;BRANCH IF NULL LINE TO DCL$RESTART
8E 7C AB D1 0147 316 CMPL PRC_L_INDCLOCK(R11),(SP)+ ;INDIRECT LEVEL CHANGE?
      E3 12 014B 317 BNEQ 35$ ;IF YES-LOOK FOR LEADING DOLLAR SIGN
      014D 318
      014D 319
      014D 320 : GET THE FIRST TOKEN FROM THE COMMAND LINE. IT SHOULD BE EITHER A VERB,
      014D 321 : A SYMBOL, OR A LABEL.
      014D 322
      FEB0' 30 014D 323 40$: BSBW DCL$MARK ;MARK CURRENT PARSE POSITION
      FEAD' 30 0150 324 BSBW DCL$GETOKEN ;GET COMMAND VERB, SYMBOL, OR LABEL
      03 12 0153 325 BNEQ 41$ ;IF EQL NO TOKEN WAS FOUND
      0123 31 0155 326 BRW NOCOMD ;SIGNAL BAD COMMAND LINE SYNTAX
      0158 327
      0158 328
      0158 329 : CHECK FOR COLON TERMINATOR. IT COULD INDICATE EITHER A LABEL OR A NULL
      0158 330 : NODE.
      0158 331
50 7E D4 0158 332 41$: CLRL -(SP) ;ASSUME TERMINATOR IS NOT A BLANK
      20 91 015A 333 CMPB #^A' ',RO ;IS TERMINATOR A BLANK?
      0D 12 015D 334 BNEQ 42$ ;NO, THEN BRANCH
      6E D6 015F 335 INCL (SP) ;SET BLANK SEEN FLAG
      FE98' 30 0161 336 SETBIT PRC_V_IND,PRC_W_FLAGS(R11) ;TEMPORARILY DISALLOW INDIRECTION
      0165 337 BSBW DCL$SETNBLK ;PEEK AT NEXT NON-BLANK
      0168 338 CLRBIT PRC_V_IND,PRC_W_FLAGS(R11) ;RESTORE INDIRECTION
50 3A 91 016C 339 42$: CMPB #^A':',RO ;LABEL OR STRING EQUATE TERMINATOR?
      10 13 016F 340 BEQL 60$ ;BRANCH IF SO
```



```

      8E D5 0171 341 TSTL (SP)+ ;WAS BLANK SEEN?
      06 13 0173 342 BEQL 43$ ;BRANCH IF NO BLANK SEEN
50    20 90 0175 343 MOVB #^A' ,RO ;RESTORE THE BLANK
      FE85' 30 0178 344 BSBW DCL$BACKUPCHAR
      005D 31 017B 345 43$: BRW SYMBOL ;ELSE, VERB OR ASSIGNMENT STATEMENT
      017E 346
      017E 347
      017E 348 : IF NULL LINE, THEN GET NEXT COMMAND.
      017E 349
      FE7F 31 017E 350 50$: BRW DCL$RESTART ;GET NEXT COMMAND
      01B1 351
      01B1 352
      01B1 353 : LABEL OR STRING ASSIGNMENT
      01B1 354
      01B1 355 : THE FIRST TOKEN WAS TERMINATED BY A ":", IF IT IS NOT IMMEDIATELY FOLLOWED
      01B1 356 : BY A SECOND COLON, I.E., IS A NULL NODE SPEC, IT MUST BE EITHER A
      01B1 357 : LABEL OR THE SYMBOL NAME IN A STRING ASSIGNMENT. WE TEMPORARILY DISALLOW
      01B1 358 : INDIRECTION WHEN PEEKING AT THE NEXT CHARACTER AFTER THE COLON, SO THAT IF
      01B1 359 : AN "a" APPEARS JUST AFTER THE LABEL, WE DON'T GET SHIFTED TO THE NEXT LEVEL
      01B1 360 : BEFORE WE GET A CHANCE TO PUT THE LABEL ENTRY IN THE RIGHT TABLE.
      01B1 361
      FE7C' 30 01B1 362 60$: BSBW DCL$MOVCHAR ;MOVE CHARACTER TO COMMAND BUFFER
      01B4 363 SETBIT PRC_V_IND,PRC_W_FLAGS(R11) ;TEMPORARILY DISALLOW INDIRECTION
      FE75' 30 01B8 364 BSBW DCL$SETCHAR ;PEEK AT NEXT CHARACTER IN INPUT BUF
      01B8 365 CLRBIT PRC_V_IND,PRC_W_FLAGS(R11) ;RESTORE INDIRECTION
50    3A 91 01BF 366 CMPB #^A7:7,RO ;COLON?
      3A 13 0192 367 BEQL NULLNODE ;IF EQL YES
      8E D5 0194 368 TSTL (SP)+ ;RESTORE THE STACK
50    3D 91 0196 369 CMPB #^A/=/,RO ;EQUAL SIGN?
      2E 13 0199 370 BEQL 80$ ;IF EQL YES
      019B 371
      019B 372
      019B 373 : LABEL
      019B 374
      019B 375 : SAVE THE LABEL IN THE SYMBOL TABLE. IF THE A LABEL OF THE SAME NAME WAS
      019B 376 : ALREADY FOUND, REPLACE IT WITH THE ONE WE HAVE JUST FOUND.
      019B 377
      019B 378
      1A 68 0B E0 019B 378 BBS #PRC_V_YLEVEL, - ;IF SET, AT CONTROL Y/C LEVEL
      50 14 AB D0 019D 379 PRC_W_FLAGS(R11),69$
      14 18 A0 02 E0 01A0 380 MOVL PRC_L_INDINPRAB(R11),RO ;DEVICE CHAR SAVED IN CURRENT RAB
      OC 18 A0 1C E1 01A4 381 BBS #DEV$V_TRM,RABSL_CTX(RO),65$ ;IF TRM, ISSUE WARNING MESSAGE
      01AE 382 BBC #DEV$V_RND,RABSL_CTX(RO),69$ ;IF NOT RANDOM DEVICE, IGNORE IT
      FE49' 30 01B4 383 DISABLE ;DISABLE CONTROL Y/C AST'S
      34 50 E9 01B7 384 BSBW DCL$ALLOC_LABEL ;ALLOCATE AND INSERT LABEL IN TABLE
      FE54 31 01BA 385 BLBC RO,130$ ;IF LBC ALLOCATION FAILURE
      F1 11 01BD 386 69$: BRW DCL$CMDSTART ;START COMMAND OVER
      01C7 387 65$: ERRMSG NOLBLS ;SIGNAL LABEL NOT ALLOWED HERE
      01C9 388 BRB 69$
      01C9 389
      01C9 390
      01C9 391 : STRING ASSIGNMENT
      01C9 392
      01C9 393 : WE HAVE DETECTED A ":@" FOLLOWING THE FIRST TOKEN ON THE LINE.
      01C9 394 : THEREFORE WE MUST BE PROCESSING A STRING ASSIGNMENT.
      FE34' 30 01C9 395
      20 11 01CC 396 80$: BSBW DCL$EQUATE ;PROCESS STRING EQUATE STATEMENT
      01CC 397 BRB 130$

```



```
01CE 398
01CE 399
01CE 400 : NULL NODE SPECIFICATION WAS FOUND. BACK UP PAST DOUBLE COLONS AND PROCESS
01CE 401 : AS A VERB.
01CE 402
01CE 403 NULLNODE:
50 FE2F' 30 01CE 404 BSBW DCL$BACKUPMOVE :BACK UP TO END OF VERB NAME
8E D5 01D1 405 TSTL (SP)+ :WAS BLANK SEEN?
06 13 01D3 406 BEQL SYMBOL :BRANCH IF NO BLANK SEEN
20 90 01D5 407 MOVW #^A',R0 :RESTORE THE BLANK
FE25' 30 01DB 408 BSBW DCL$BACKUPCHAR :
01DB 409
01DB 410 :
01DB 411 : COMMAND VERB, INTEGER ASSIGNMENT, SUBSTRING ASSIGNMENT, OR BITFIELD ASSIGNMENT
01DB 412 :
01DB 413 : THE FIRST TOKEN WAS NOT TERMINATED BY A ":", IT MUST BE EITHER A COMMAND VERB
01DB 414 : OR THE SYMBOL NAME IN AN INTEGER, SUBSTRING, OR BITFIELD ASSIGNMENT.
01DB 415 :
01DB 416 : PROCESS INTEGER, SUBSTRING, OR BITFIELD ASSIGNMENT
01DB 417
50 5B 8F 91 01DB 418 SYMBOL: CMPB #^A/[/,R0 :SUBSTRING OF BIT FIELD SUBSTRING ASSIGNMENT
05 12 01DF 419 BNEQ 120$ :IF NEQ NO
FE1C' 30 01E1 420 BSBW DCL$SUBASSIGN :PROCESS SUBSTRING OR BIT FIELD ASSIGNMENT
08 11 01E4 421 BRB 130$ :
50 3D 91 01E6 422 120$: CMPB #^A/=/,R0 :SYMBOL ASSIGNMENT?
06 12 01E9 423 BNEQ COMMAND :IF NEQ NO
FE12' 30 01EB 424 BSBW DCL$EVALUATE :EVALUATE ARITHMETIC ASSIGNMENT STATEMENT
0146 31 01EE 425 130$: BRW ERROR_EXIT :
01F1 426
01F1 427 :
01F1 428 : COMMAND VERB
01F1 429 :
01F1 430 : WE HAVE DETERMINED THAT WE ARE PARSING A COMMAND THAT BEGINS WITH A VERB.
01F1 431 : TRANSLATE THE FIRST TOKEN ON THE COMMAND LINE (IF POSSIBLE).
01F1 432 :
54 06 BB 01F1 433 COMMAND:PUSHR #^M<R1,R2> :SAVE COMMAND VERB PARAMETERS
FE0A' 30 01F3 434 BSBW DCL$SYN_STRING :SEARCH FOR SYMBOL DEFINITION
51 7D 01F6 435 MOVQ R1,R4 :SAVE VALUE PARAMETERS
06 BA 01F9 436 POPR #^M<R1,R2> :RESTORE COMMAND VERB PARAMETERS
54 D5 01FB 437 TSTL R4 :SYMBOL DEFINED AND HAVE VALUE?
03 12 01FD 438 BNEQ 135$ :IF NEQ YES
0083 31 01FF 439 BRW PRCVERB :IF NOT SYMBOL, SEARCH VERB TABLE
0202 440
0202 441 :
0202 442 : THE FIRST TOKEN IS A SYMBOL. GET ITS VALUE.
0202 443 :
F486 CA F492 CA 9E 0202 444 135$: MOVAB WRK_G_BUFFER(R10),WRK_L_EXPANDPTR(R10) :RESET EXPANSION BUFFER POINT
56 F48E CA 01 C1 0209 445 ADDL3 #1,WRK_L_CHARPTR(R10),R6 :GET CHARACTER POINTER
66 20 91 020F 446 CMPB #^A/'',(R6) :ALREADY A BLANK HERE?
03 13 0212 447 BEQL 140$ :IF EQL YES
76 20 90 0214 448 MOVW #^A/'',-(R6) :ELSE INSERT A TRAILING BLANK
FF A544 27 91 0217 449 140$: CMPB #^A/'',-1(R5)[R4] :DOES THE SYMBOL END WITH '?
04 12 021C 450 BNEQ 150$ :IF NEQ NO
54 D7 021E 451 DECL R4 :ELSE REMOVE THE ' FROM THE SYMBOL
56 D6 0220 452 INCL R6 :AND REMOVE THE ADDED TRAILING BLANK
54 C2 0222 453 150$: SUBL R4,R6 :BACK UP OVER SYMBOL'S LENGTH
F48E CA 56 01 C3 0225 454 SUBL3 #1,R6,WRK_L_CHARPTR(R10) :STORE NEW CHARACTER POINTER
```

```
66 65 54 28 022B 455      MOVCL R4,(R5),(R6)      :MOVE TRANSLATED COMMAND TO INPUT BUFFER
                                022F 456      SETBIT  PRC_V_IND,PRC_W_FLAGS(R11) :TEMPORARILY DISALLOW INDIRECTION
                                FDCA' 30 0233 457      BSBW   DCL$SETNBLK      :SKIP OVER LEADING BLANKS
                                0236 458      CLRBIT  PRC_V_IND,PRC_W_FLAGS(R11) :ENABLE INDIRECTION
                                50      95 023A 459      TSTB   R0              :COMMAND NOP'D?
                                03      12 023C 460      BNEQ   152$           :BR IF NOT
                                FDBF' 31 023E 461      BRW    DCL$RESTART      :
                                50 40 8F 91 0241 462 152$: CMPB   #^A/^/,R0      :PROCEDURE FILE?
                                03      12 0245 463      BNEQ   155$           :IF NEQ NO
                                FDC7' 31 0247 464      BRW    DCL$CMDSTART      :READ FROM THE COMMAND FILE
                                024A 465
                                024A 466
                                024A 467      : IF THE SYMBOL DEFINES A FOREIGN COMMAND, THEN PROCESS IT AS SUCH.
                                024A 468      : OTHERWISE, SEARCH THE VERB TABLE FOR THE VERB.
                                024A 469
                                FDB3' 30 024A 470 155$: BSBW   DCL$MARK          :MARK CURRENT PARSE POSITION
                                FDB0' 30 024D 471      BSBW   DCL$SETNBLK      :POINT TO NEXT NONBLANK CHARACTER
                                50 24 91 0250 472      CMPB   #^A/^/,R0      :DOLLAR SIGN?
                                21 12 0253 473      BNEQ   CMDVERB          :IF NEQ NO
                                0255 474
                                0255 475      .DSABL  LSB
                                0255 476
                                0255 477
                                0255 478      : FOREIGN COMMAND
                                0255 479
                                0255 480      : PROCESS THE FILE SPECIFICATION AND THEN PROCESS THE REST OF THE COMMAND
                                0255 481      : LINE.
                                0255 482
                                0255 483      FORVERB:
                                53 03 9A 0255 484      MOVZBL  #PTR K PARAMETR,R3 :COME HERE FOR FOREIGN VERBS
                                FDA5' 30 0258 485      BSBW   DCL$PROCFIL      :SET TOKEN CONTEXT OF FILESPEC
                                24 50 E9 0258 486      BLBC   R0,ERROR          :PROCESS FILE SPECIFICATION
                                BA AA 0C C2 025E 487      SUBL   #PTR C LENGTH,WRK_L_RSLNXT(R10) :IF LBC FILE SPECIFICATION ERROR
                                7E 51 7D 0262 488      MOVQ   R1,-(SP)          :UNDO RESULT TOKEN DESCRIPTOR
                                0347 30 0265 489      BSBW   PROCFORM          :SAVE FILENAME DESCRIPTOR PARAMETERS
                                C2 AA 00'8F 90 0268 490      MOVB   #CLISK_VERB_FORE,WRK_B_VERBTYP(R10) :PROCESS FOREIGN COMMAND
                                06 BA 026D 491      POPR    #^M<R1,R2>          :SET FOREIGN COMMAND
                                53 0E'8F 9A 026F 492      MOVZBL #IMG K EXTIMAGE&*X7F,R3 :RESTORE FILENAME DESCRIPTOR PARAMETERS
                                007A 31 0273 493      BRW    EXECEXT          :SET EXTERNAL IMAGE INDEX
                                0276 494
                                0276 495
                                0276 496      : GET THE TRANSLATED COMMAND VERB
                                0276 497
                                0276 498      CMDVERB:
                                FD87' 30 0276 499      BSBW   DCL$GETOKEN      :GET COMMAND VERB TOKEN
                                OA 12 0279 500      BNEQ   PRCVERB          :PROCEED IF VALID
                                00B2 31 027B 501      NOCMD: STATUS NOCMD          :ERROR IF NULL LINE
                                0282 502      ERROR: BRW    ERROR_EXIT
                                0285 503
                                0285 504
                                0285 505      : LOOK THE VERB UP IN THE COMMAND TABLES.
                                0285 506
                                0285 507      PRCVERB:
                                09 68 AB 08 E1 0285 508      BBC    #PRC_V_YLEVEL,PRC_W_FLAGS(R11),10$ :BR IF NOT AT ^Y LEVEL
                                04 00 BB 18 E1 028A 509      BBC    #PSLV-CURMOD,@PRC_C_SAVAP(R11),10$ :IF CLR PREVIOUS MODE SUPER
                                028F 510      SETBIT  PRC_V_IND,PRC_W_FLAGS(R11) :DISABLE ^@ FILE RECOGNITION
                                58 00000000'GF D0 0293 511 10$: MOVL  G^CTL$AG_CLITABCE,R8 :GET ADDRESS OF DATA BASE VECTOR
```

```
FD63' 30 029A 512 BSBW DCL$SEARCH_VERB ;SEARCH VERB TABLE FOR VERB
E2 50 E9 029D 513 BLBC RO,ERROR ;BRANCH IF ERROR
029A 514
029A 515
029A 516
029A 517
029A 518
029A 519
029A 520
029A 521
029A 522
029A 523
02 524
05 04 A8 E1 02A0 524 BBC #CMD V FOREIGN, - ;BRANCH IF NOT FOREIGN
0307 30 02A2 525 CMD W FLAGS(R8),20$
26 11 02A5 526 BSBW PROCFORM ;PROCESS FOREIGN COMMAND
02AA 527 BRB BUILD_IMAGE ;PROCESS THE IMAGE NAME
OF 04 A8 03 E1 02AA 529 20$: BBC #CMD V IMMED,CMD W FLAGS(R8),30$ ;BR IF NOT IMMEDIATE
52 E2 AA D0 02AF 530 MOVL WRK_L_IMAGE(R10),R2 ;GET ADDR OF ASCIC ROUTINE NAME
51 82 9A 02B3 531 MOVZBL (R2)+,R1 ;GET COUNT IN R1, ADDRESS IN R2
0336 30 02B6 532 BSBW DCL$LOCATE_INTERNAL ;LOCATE INTERNAL ROUTINE INDEX
C6 50 E9 02B9 533 BLBC RO,ERROR ;BRANCH IF ERROR
3C 11 02BC 534 BRB IMMED ;DISPATCH THE IMMEDIATE COMMAND
02BE 535
FD3F' 30 02BE 536 30$: BSBW DCL$PARSE_COMMAND ;PARSE THE COMMAND QUALS AND PARMS
000380B0 8F 50 D1 02C1 537 CMPL RO,#CLIS_NOCMD ;IF CTRL/Z WAS ENTERED,
03 12 02C8 538 BNEQ 40$ ;THEN GET A NEW COMMAND
FD33 31 02CA 539 BRW DCL$RESTART
B2 50 E9 02CD 540 40$: BLBC RO,ERROR ;SIGNAL ANY SYNTAX ERRORS
02D0 541
02D0 542
02D0 543
02D0 544
02D0 545
02D0 546
52 E2 AA D0 02D0 546 BUILD_IMAGE: MOVL WRK_L_IMAGE(R10),R2 ;GET IMAGE/ROUTINE ASCIC NAME ADDRESS
51 82 9A 02D4 547 MOVZBL (R2)+,R1 ;GET COUNT IN R1, ADDRESS IN R2
53 0E'8F 9A 02D7 548 MOVZBL #IMG_K_EXTIMAGE&^X7F,R3 ;ASSUME EXTERNAL IMAGE
00 E0 02DB 549 BBS #WRK_V_CLIRTN, - ;BRANCH IF INTERNAL ROUTINE
02 F2 AA 02DD 550 WRK W FLAGS2(R10),5$
0E 11 02E0 551 BRB EXECEXT ;EXECUTE EXTERNAL COMMAND
030A 30 02E2 552 5$: BSBW DCL$LOCATE_INTERNAL ;LOCATE INTERNAL ROUTINE INDEX
9A 50 E9 02E5 553 BLBC RO,ERROR ;BRANCH IF ERROR
02E8 554
02E8 555
02E8 556
02E8 557
02E8 558
02E8 559
53 DD 02E8 559 PUSH R3 ;SAVE COMMAND INDEX
FD13' 30 02EA 560 BSBW DCL$SORT_TOKENS ;SORT DESCRIPTOR TABLE
53 8ED0 02ED 561 POPL R3 ;RESTORE COMMAND INDEX
02F0 562
02F0 563
02F0 564
BA AA F9B6 CA 9E 02F4 565 SETBIT WRK_V_COMMAND,WRK_W_FLAGS(R10) ;SET COMMAND EXECUTION IN PROGRESS
02FA 566 MOVAB WRK_G_RESULT(R10),WRK_L_RSLNXT(R10) ;RESET NEXT TOKEN TO RETRIEVE
02FA 567
02FA 568 ; INTERNAL COMMAND PARAMETERS:
```



```
02FA 569 :  
02FA 570 : R1 = LENGTH OF IMAGE FILENAME (IF ANY).  
02FA 571 : R2 = ADDRESS OF IMAGE FILENAME (IF ANY).  
02FA 572 :  
02FA 573 : R8 AND R9 ARE SET UP AFTER THE CONDITIONAL RUNDOWN CALL BELOW  
02FA 574 :  
02FA 575 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.  
02FA 576 : R9 = ADDRESS OF SCRATCH STACK.  
02FA 577 : R10 = BASE ADDRESS OF COMMAND WORK AREA.  
02FA 578 : R11 = BASE ADDRESS OF PROCESS WORK AREA.  
02FA 579 :  
02FA 580 : IMMEDIATE COMMAND PARAMETERS:  
02FA 581 :  
02FA 582 : R10 = BASE ADDRESS OF COMMAND WORK AREA.  
02FA 583 : R11 = BASE ADDRESS OF PROCESS WORK AREA.  
02FA 584 :  
02FA 585 :  
23 00AF CB 04 E1 02FA 586 IMMED: :IMMEDIATE COMMAND EXECUTION  
02FA 587 BBC #PRC_V_PRIV,PRC_B_FLAGS2(R11),10$ ;BR IF UNPRIVILEGED IMAGE  
0300 588 :  
0300 589 :  
0300 590 : NEED TO RUN DOWN PRIVILEGED IMAGE, BUT ALLOW THE FOLLOWING COMMANDS  
0300 591 :  
53 03'8F 91 0300 592 CMPB #IMG_K_CONTINUE & ^X7F,R3 ;CONTINUE COMMAND?  
1D 13 0304 593 BEQL 10$ ;BR IF YES  
53 28'8F 91 0306 594 CMPB #IMG_K_SPAWN & ^X7F,R3 ;SPAWN COMMAND?  
17 13 030A 595 BEQL 10$ ;BR IF YES  
53 29'8F 91 030C 596 CMPB #IMG_K_ATTACH & ^X7F,R3 ;ATTACH COMMAND?  
11 13 0310 597 BEQL 10$ ;BR IF YES  
05 FO AA 01 E3 0312 598 BBBS #WRK_V_COMMAND,WRK_W_FLAGS(R10),2$  
FCE6' 30 0317 599 BSBW DCL$FORCEXIT ;RUN DOWN THE PRIVILEGED IMAGE  
07 11 031A 600 BRB 10$ ;  
FCE1' 30 031C 601 :  
031C 602 2$: BSBW DCL$FORCEXIT ;RUN DOWN THE PRIVILEGED IMAGE  
031F 603 CLRBIT WRK_V_COMMAND,WRK_W_FLAGS(R10)  
0323 604 :  
0323 605 :  
0323 606 : SETUP SCRATCH STORAGE FOR USE BY INTERNAL ROUTINES  
0323 607 :  
5E 59 5E D0 0323 608 10$: MOVL SP,R9 ;SET ADDRESS OF SCRATCH STACK  
FE00 CE 9E 0326 609 MOVAB -WRK_C_SCRSTKSIZ(SP),SP ;ALLOCATE SCRATCH AREA BEFORE STACK  
6E 9F 032B 610 PUSHAB (SP) ;BUILD SCRATCH BUFFER DESCRIPTOR  
7E 0200 8F 3C 032D 611 MOVZWL #WRK_C_SCRSTKSIZ,-(SP)  
58 5E D0 0332 612 MOVL SP,R8 ;SET ADDRESS OF SCRATCH BUFFER DESCRIPTOR  
12 10 0335 613 BSBB IMAGECASE ;EXECUTE INTERNAL IMAGE  
0337 614 :  
0337 615 :  
0337 616 : THE FOLLOWING CODE IS DUPLICATED IN IMAGEEXEC FOR EXTERNAL IMAGES  
0337 617 : BECAUSE IT MUST BE DONE BEFORE IMAGE RUNDOWN, NOT AFTERWARDS.  
0337 618 :  
0337 619 ERROR_EXIT: :ERROR EXIT  
03 50 E8 0337 620 BLBS R0,10$ ;IF LBS SUCCESSFUL COMPLETION  
FCC3' 30 033A 621 BSBW DCL$ERRORMSG ;OUTPUT SYSTEM ERROR MESSAGE  
00000000'EF 16 033D 622 10$: JSB DCL$SET STATUS ;SET COMPLETION STATUS  
023A 30 0343 623 BSBW DCL$FLUSH ;FLUSH COMMAND BUFFER  
FCB7 31 0346 624 BRW DCL$RESTART ;  
0349 625 :
```

35'8F 00 53 8F

```

0349 627 .MACRO INTIMAGE NAME
0349 628 .WORD XXX$'NAME'-10$
0349 629 IMG_K 'NAME' = $INTIMAGES
0349 630 $INTIMAGES = $INTIMAGES + 1
0349 631 .ENDM
0349 632
0349 633 IMAGECASE:
0349 634 CASEB R3,#0,#<90$-10$>/2-1
034E 635 10$:
034E 636 .LIST MEB
034E 637 INTIMAGES
0072' 034E .WORD XXX$ALLOCATE-10$
0078' 0350 .WORD XXX$ASSIGN-10$
007E' 0352 .WORD XXX$CLOSE-10$
0084' 0354 .WORD XXX$CONTINUE-10$
008A' 0356 .WORD XXX$DEALLOCAT-10$
0090' 0358 .WORD XXX$DEASSIGN-10$
0096' 035A .WORD XXX$DEBUG-10$
009C' 035C .WORD XXX$DECK-10$
00A2' 035E .WORD XXX$DEFINE-10$
00A8' 0360 .WORD XXX$DEPOSIT-10$
00AE' 0362 .WORD XXX$DELSYM-10$
00B4' 0364 .WORD XXX$EOD-10$
00BA' 0366 .WORD XXX$EXAMINE-10$
00C0' 0368 .WORD XXX$EXIT-10$
00C6' 036A .WORD XXX$EXTIMAGE-10$
00CC' 036C .WORD XXX$GOTO-10$
00D2' 036E .WORD XXX$IF-10$
00D8' 0370 .WORD XXX$INQUIRE-10$
00DE' 0372 .WORD XXX$LOGOUT-10$
00E4' 0374 .WORD XXX$MCR-10$
00EA' 0376 .WORD XXX$ON-10$
00F0' 0378 .WORD XXX$OPEN-10$
00F6' 037A .WORD XXX$READ-10$
00FC' 037C .WORD XXX$RUN-10$
0102' 037E .WORD XXX$SETCTLY-10$
0108' 0380 .WORD XXX$SETDEFAULT-10$
010E' 0382 .WORD XXX$SETON-10$
0114' 0384 .WORD XXX$SETPROT-10$
011A' 0386 .WORD XXX$SETUIC-10$
0120' 0388 .WORD XXX$SETVERIFY-10$
0126' 038A .WORD XXX$SHOWDEF-10$
012C' 038C .WORD XXX$SHOWPROT-10$
0132' 038E .WORD XXX$SHOWQUOTA-10$
0138' 0390 .WORD XXX$SHOWSTAT-10$
013E' 0392 .WORD XXX$SHOWSYMBL-10$
0144' 0394 .WORD XXX$SHOWTIME-10$
014A' 0396 .WORD XXX$SHOWTRAN-10$
0150' 0398 .WORD XXX$STOP-10$
0156' 039A .WORD XXX$WAIT-10$
015C' 039C .WORD XXX$WRITE-10$
0162' 039E .WORD XXX$SPAWN-10$
0168' 03A0 .WORD XXX$ATTACH-10$
016E' 03A2 .WORD XXX$EOJ-10$
0174' 03A4 .WORD XXX$CANCEL-10$
017A' 03A6 .WORD XXX$SETPROMPT-10$
0180' 03A8 .WORD XXX$RECALL-10$

```

:BASE ADDRESS OF CASE TABLE

:GENERATE INTERNAL IMAGE CASE TABLE

```
0186' 03AA .WORD XXX$DEFKEY-10$
018C' 03AC .WORD XXX$SHOWKEY-10$
0192' 03AE .WORD XXX$DELKEY-10$
0198' 03B0 .WORD XXX$SETKEY-10$
019E' 03B2 .WORD XXX$SETFLUSH-10$
01A4' 03B4 .WORD XXX$CRETABLE-10$
01AA' 03B6 .WORD XXX$CONNECT-10$
01B0' 03B8 .WORD XXX$DISCONNECT-10$
      03BA .NLIST MEB
50 0000'8F 3C 03BA 638 90$: MOVZWL #SS$_ILLSER,RO ;SET SERVICE ERROR
      05 03BF 639 RSB
      03C0 640
      03C0 641
      03C0 642
      03C0 643 .MACRO INTIMAGE NAME
      03C0 644 XXX$'NAME': JMP DCL$'NAME'
      03C0 645 IMG K 'NAME' = $INTIMAGES
      03C0 646 $INTIMAGES = $INTIMAGES + 1
      03C0 647 .ENDM
      03C0 648
      03C0 649 .LIST MEB
      03C0 650 INTIMAGES ;GENERATE INTERNAL IMAGE CASE TABLE
00000000'EF 17 03C0 XXX$ALLOCATE: JMP DCL$ALLOCATE
00000000'EF 17 03C6 XXX$ASSIGN: JMP DCL$ASSIGN
00000000'EF 17 03CC XXX$CLOSE: JMP DCL$CLOSE
00000000'EF 17 03D2 XXX$CONTINUE: JMP DCL$CONTINUE
00000000'EF 17 03D8 XXX$DEALLOCAT: JMP DCL$DEALLOCAT
00000000'EF 17 03DE XXX$DEASSIGN: JMP DCL$DEASSIGN
00000000'EF 17 03E4 XXX$DEBUG: JMP DCL$DEBUG
00000547'EF 17 03EA XXX$DECK: JMP DCL$DECK
00000000'EF 17 03F0 XXX$DEFINE: JMP DCL$DEFINE
00000000'EF 17 03F6 XXX$DEPOSIT: JMP DCL$DEPOSIT
00000000'EF 17 03FC XXX$DELSYM: JMP DCL$DELSYM
0000053A'EF 17 0402 XXX$EOD: JMP DCL$EOD
00000000'EF 17 0408 XXX$EXAMINE: JMP DCL$EXAMINE
00000000'EF 17 040E XXX$EXIT: JMP DCL$EXIT
00000000'EF 17 0414 XXX$EXTIMAGE: JMP DCL$EXTIMAGE
00000000'EF 17 041A XXX$GOTO: JMP DCL$GOTO
00000000'EF 17 0420 XXX$IF: JMP DCL$IF
00000000'EF 17 0426 XXX$INQUIRE: JMP DCL$INQUIRE
00000000'EF 17 042C XXX$LOGOUT: JMP DCL$LOGOUT
00000000'EF 17 0432 XXX$MCR: JMP DCL$MCR
00000000'EF 17 0438 XXX$ON: JMP DCL$ON
00000000'EF 17 043E XXX$OPEN: JMP DCL$OPEN
00000000'EF 17 0444 XXX$READ: JMP DCL$READ
00000000'EF 17 044A XXX$RUN: JMP DCL$RUN
00000000'EF 17 0450 XXX$SETCTLY: JMP DCL$SETCTLY
00000000'EF 17 0456 XXX$SETDEFAULT: JMP DCL$SETDEFAULT
00000000'EF 17 045C XXX$SETON: JMP DCL$SETON
00000000'EF 17 0462 XXX$SETPROT: JMP DCL$SETPROT
00000000'EF 17 0468 XXX$SETUIC: JMP DCL$SETUIC
00000000'EF 17 046E XXX$SETVERIFY: JMP DCL$SETVERIFY
00000000'EF 17 0474 XXX$SHOWDEF: JMP DCL$SHOWDEF
00000000'EF 17 047A XXX$SHOWPROT: JMP DCL$SHOWPROT
00000000'EF 17 0480 XXX$SHOWQUOTA: JMP DCL$SHOWQUOTA
00000000'EF 17 0486 XXX$SHOWSTAT: JMP DCL$SHOWSTAT
00000000'EF 17 048C XXX$SHOWSYMBL: JMP DCL$SHOWSYMBL
00000000'EF 17 0492 XXX$SHOWTIME: JMP DCL$SHOWTIME
```


00000000'EF	17	0498	XXX\$SHOWTRAN:	JMP	DCL\$SHOWTRAN
00000000'EF	17	049E	XXX\$STOP:	JMP	DCL\$STOP
00000000'EF	17	04A4	XXX\$WAIT:	JMP	DCL\$WAIT
00000000'EF	17	04AA	XXX\$WRITE:	JMP	DCL\$WRITE
00000000'EF	17	04B0	XXX\$SPAWN:	JMP	DCL\$SPAWN
00000000'EF	17	04B6	XXX\$ATTACH:	JMP	DCL\$ATTACH
00000504'EF	17	04BC	XXX\$EOJ:	JMP	DCL\$EOJ
00000000'EF	17	04C2	XXX\$CANCEL:	JMP	DCL\$CANCEL
00000000'EF	17	04C8	XXX\$SETPROMPT:	JMP	DCL\$SETPROMPT
00000000'EF	17	04CE	XXX\$RECALL:	JMP	DCL\$RECALL
00000000'EF	17	04D4	XXX\$DEFKEY:	JMP	DCL\$DEFKEY
00000000'EF	17	04DA	XXX\$SHOWKEY:	JMP	DCL\$SHOWKEY
00000000'EF	17	04E0	XXX\$DELKEY:	JMP	DCL\$DELKEY
00000000'EF	17	04E6	XXX\$SETKEY:	JMP	DCL\$SETKEY
00000000'EF	17	04EC	XXX\$SETFLUSH:	JMP	DCL\$SETFLUSH
00000000'EF	17	04F2	XXX\$CRETABLE:	JMP	DCL\$CRETABLE
00000000'EF	17	04F8	XXX\$CONNECT:	JMP	DCL\$CONNECT
00000000'EF	17	04FE	XXX\$DISCONNECT:	JMP	DCL\$DISCONNECT
		0504	651 .NLIST	MEB	

```

0504 653 .SBTTL CALL LOGINOUT TO ABORT THE PROCESS
0504 654 -----
0504 655 : DCL$ABORT
0504 656 :
0504 657 : CONTROL IS TRANSFERRED HERE TO ABORT THE PROCESS BY CALLING
0504 658 : THE LOGINOUT IMAGE. CARE IS TAKEN TO SETUP THE RESULT
0504 659 : PARSE DESCRIPTORS SO THAT LOGINOUT CAN USE RESULT PARSING
0504 660 : AS IF LOGOUT WAS GIVEN AS A COMMAND LINE.
0504 661 :
0504 662 : INPUTS:
0504 663 :
0504 664 : NONE
0504 665 : -----
0504 666 :
0504 667 DCL$EOJ::
0504 668 DCL$ABORT::
0504 669 MOVAB WRK_G_RESULT(R10),- ;RESET RESULT ARRAY PTR
0508 670 WRK_L_RSLNXT(R10) ;
050A 671 MOVAB WRK_G_BUFFER(R10),R2 ;GET ADDRESS OF EXPANSION BUFFER
050F 672 MOVL R2,WRK_L_EXPANDPTR(R10) ;RESET EXPANSION PTR
0514 673 MOVL #^A'LOGO',(R2) ;STORE VERB INTO BUFFER
051B 674 MOVL #4,R1 ;SET LENGTH OF VERB
051E 675 MOVAB LOGOUT,WRK_L_IMAGE(R10) ;SET ROUTINE IN CASE NOT FOUND
0526 676 SETBIT WRK_V_CLIRTN,WRK_W_FLAGS2(R10)
052A 677 MOVL G^CTL$AG CLITABLE,R8 ;GET ADDRESS OF DATA BASE VECTOR
0531 678 BSBW DCL$SEARCH_VERB ;LOCATE COMMAND DEFINITION FOR
0534 679 ;LOGOUT CLI INTERFACE (IGNORE ERRORS)
0534 680 BSBW DCL$GENEOL ;CREATE AN EOL TOKEN
0537 681 BRW BUILD_IMAGE ;PROCESS LOGOUT COMMAND

```

```

053A 683 .SBTTL EOD/DECK COMMANDS
053A 684 :+
053A 685 : DCL$EOD/DCL$DECK
053A 686 :
053A 687 : THESE COMMANDS ARE PROCESSED BY RMS AND SHOULD NOT BE SEEN BY THE CLI.
053A 688 : RMS PASSES THEM THROUGH IF THE SYNTAX OR VALUE IS BAD. ONE CASE IS
053A 689 : NOT TREATED AS AN ERROR, THAT IS "EOD" ALONE ON A LINE. THIS COULD
053A 690 : HAPPEN IF THE STATEMENT WAS ENTERED WITHOUT AN IMAGE RUN.
053A 691 :-
053A 692 :
053A 693 DCL$EOD::
053A 694 BSBW DCL$SETNBLK ;LOOK FOR NEXT CHARATER IN LINE
053D 695 BNEQ DCL$DECK ;BR IF NOT A NULL LINE
053F 696 STATUS NORMAL ;SET OK STATUS
05 0546 697 RSB ;
0547 698 :
0547 699 DCL$DECK::
0547 700 STATUS IVVALU ; ASSUME ERROR WAS INVALID VALUE
05 054E 701 RSB

```



```

054F 703 .SBTTL CHECK FOR CONTROL Y/C AST PENDING
054F 704 :+
054F 705 DCL$CHECK_AST - CHECK FOR CONTROL Y/C AST PENDING
054F 706 :
054F 707 THIS ROUTINE IS CALLED TO CHECK FOR A PENDING CONTROL Y/C AST.
054F 708 :
054F 709 INPUTS:
054F 710 :
054F 711 NONE.
054F 712 :
054F 713 OUTPUTS:
054F 714 :
054F 715 IF A CONTROL Y/C AST IS PENDING, THEN CONTROL IS TRANSFERED TO THE COMMAND
054F 716 RESTART ENTRY POINT. OTHERWISE CONTROL IS RETURNED TO THE CALLER.
054F 717 :-
054F 718 :
054F 719 .ENABL LSB
054F 720 DCL$CHECK_AST:: :CHECK FOR PENDING CONTROL Y/C AST
2B 6B AB 01 E1 054F 721 BBC #PRC V CNTRLY,PRC_W_FLAGS(R11),50$ ;IF CLR, NO CONTROL Y/C REQUEST
FAA9 31 0554 722 10$: BRW DCL$RESTART ;

```

```

0557 724 .SBTTL ENABLE/DISABLE CONTROL Y/C AST'S
0557 725 :+
0557 726 :DCL$DISABLE - DISABLE CONTROL Y/C AST'S
0557 727 :
0557 728 :THIS ROUTINE IS CALLED TO DISABLE/ENABLE CONTROL Y/C AST'S.
0557 729 :
0557 730 :INPUTS:
0557 731 :
0557 732 :NONE.
0557 733 :
0557 734 :OUTPUTS:
0557 735 :
0557 736 :CONTROL Y/C AST'S ARE DISABLED, THE PREVIOUS ENABLE STATE IS SAVED, AND
0557 737 :A CO-ROUTINE CALL IS MADE BACK TO THE CALLER. WHEN CONTROL RETURNS, THE
0557 738 :AST STATE IS RESTORED TO ITS VALUE BEFORE THE DISABLE.
0557 739 :
0557 740 :ALL REGISTERS ARE PRESERVED ACROSS CALL
0557 741 :-
0557 742 :
0557 743 DCL$DISABLE::
0557 744 :DISABLE CONTROL Y/C AST'S
0557 745 :PUSHL (SP) :COPY RETURN ADDRESS
0559 746 :MOVZWL PRC_W_FLAGS(R11),4(SP) :SAVE PREVIOUS DISABLE STATE
055E 747 :SETBIT PRC_V_DISABL,PRC_W_FLAGS(R11) :DISABLE CONTROL Y/C AST'S
0562 748 :JSB 0(SP)+ :CALL THE CALLER BACK
13 04 AE 02 E0 0564 749 :BBS #PRC_V_DISABL,4(SP),40$ :IF SET, AST'S PREVIOUSLY DISABLED
05 68 AB 0B E0 0569 750 :BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),20$ :IF SET, AT CONTROL Y/C LEVEL
5C AB D5 056E 751 :TSTL PRC_C_INDEPTH(R11) :INDIRECT LEVEL ZERO?
05 05 12 0571 752 :BNEQ 30$ :IF NEQ NO
DC 68 AB 01 E0 0573 753 20$: BBS #PRC_V_CNTRL,PRC_W_FLAGS(R11),10$ :IF SET, CONTROL Y/C REQUEST
0578 754 30$: CLRBIT PRC_V_DISABL,PRC_W_FLAGS(R11) :ENABLE CONTROL Y/C AST'S
6E 8ED0 057C 755 40$: POPL (SP) :REMOVE PREVIOUS STATE FROM STACK
057F 756 50$: RSB :
0580 756 :.DSABL LSB

```

```

0580 758 .SBTTL FLUSH COMMAND BUFFER
0580 759 :+
0580 760 DCL$FLUSH - FLUSH COMMAND BUFFER
0580 761 :
0580 762 THIS ROUTINE IS CALLED TO READ CHARACTERS FROM THE COMMAND BUFFER UNTIL AN
0580 763 END OF LINE IS ENCOUNTERED.
0580 764 :
0580 765 INPUTS:
0580 766 :
0580 767 NONE.
0580 768 :
0580 769 OUTPUTS:
0580 770 :
0580 771 INDIRECT FILE RECOGNITION IS DISABLED AND CHARACTERS ARE READ FROM THE
0580 772 INPUT BUFFER UNTIL AN END OF LINE IS ENCOUNTERED.
0580 773 :-
0580 774 :
0580 775 DCL$FLUSH:::FLUSH COMMAND BUFFER
29 68 AB 0B E0 0580 776 BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),20$ : IF AT ^Y LEVEL DON'T FLUSH
50 14 AB D0 0585 777 MOVL PRC_C_INDINPRAB(RT1),R0 : GET CURRENT INPUT RAB
20 18 A0 02 E0 0589 778 BBS #DEV$V TRM,RAB$C_CTX(R0),20$ : LIKEWISE FOR TERMINALS
F486 CA F492 CA 9E 058E 779 MOVAB WRK_G_BUFFER(R10),WRK_L_EXPANDPTR(R10) : SET BUFFER TO RECEIVE ANY CO
0595 780 SETBIT PRC_V_IND,PRC_W_FLAGS(R11) : DISABLE INDIRECT FILE RECOGNITION
0599 781 SETBIT PRC_V_FLUSH,PRC_B_FLAGS2(R11) : SET FLUSH IN PROGRESS
FA5E' 30 059F 782 10$: BSBW DCL$GETCHAR : GET CHARACTER FROM INPUT BUFFER
FB 12 05A2 783 BNEQ 10$ : IF NEQ NOT END OF LINE
05A4 784 CLRBIT PRC_V_FLUSH,PRC_B_FLAGS2(R11) : RESET FLUSH IN PROGRESS FLAG
05AA 785 CLRBIT PRC_V_IND,PRC_W_FLAGS(R11) : ENABLE INDIRECT FILE RECOGNITION
05 05AE 786 20$: RSB ;

```

```
05AF 788 .SBTTL PROCESS FOREIGN COMMAND
05AF 789 :---
05AF 790 PROCFORM - GENERATE RESULT DESCRIPTOR FOR FOREIGN COMMAND LINE
05AF 791 :
05AF 792 PROCESS FOREIGN COMMAND AND BUILD TOKEN DESCRIPTOR
05AF 793 FOR THE REST OF THE LINE AFTER THE VERB.
05AF 794 :
05AF 795 INPUTS:
05AF 796 :
05AF 797 NONE
05AF 798 :
05AF 799 OUTPUTS:
05AF 800 :
05AF 801 A TOKEN DESCRIPTOR IS GENERATED FOR THE REST OF THE LINE AND
05AF 802 A TOKEN DESCRIPTOR IS GENERATED FOR THE END OF LINE.
05AF 803 :---
05AF 804 :
05AF 805 :
05AF 806 PROCFORM:
05AF 807 SETBIT PRC_V_IND,PRC_W_FLAGS(R1) :PROCESS FOREIGN COMMAND
20 FA4A' 30 05B3 808 BSBW DCL$SETCHAR :DISABLE INDIRECT FILE INTERPRETATION
50 91 05B6 809 CMPB R0,#^A' :PEEK AT NEXT CHARACTER IN INPUT BUFFER
03 12 05B9 810 BNEQ 5$ :BLANK?
FA42' 30 05BB 811 BSBW DCL$MOVCHAR :BRANCH IF NO LEADING BLANK
FA3F' 30 05BE 812 5$: BSBW DCL$MARK :MOVE BLANK BEFORE MARK
FA3C' 30 05C1 813 10$: BSBW DCL$MOVCHAR :MARK POSITION AFTER FILESPEC & BLANK
FB 12 05C4 814 BNEQ 10$ :MOVE CHARACTER TO COMMAND BUFFER
F486 CA D7 05C6 815 DECL WRK_L_EXPANDPTR(R10) :UNTIL END OF LINE
FA33' 30 05CA 816 BSBW DCL$MARKEDTOKEN :BACKUP OVER EOL CHARACTER
57 51 7D 05CD 817 MOVQ R1,R7 :GET DESCRIPTOR OF REST OF LINE
54 D4 05D0 818 CLRL R4 :COPY DESCRIPTOR
56 D4 05D2 819 CLRL R6 :CLEAR ITEM NUMBER
59 D4 05D4 820 CLRL R9 :CLEAR FLAGS
55 03 9A 05D6 821 MOVZBL #PTR_K_PARAMETR,R5 :CLEAR ENTITY BLOCK ADDRESS
FA24' 30 05D9 822 BSBW DCL$GENDESCR :SET ITEM TYPE TO PARAMETER
55 04 9A 05DC 823 MOVZBL #PTR_K_ENDLINE,R5 :GENERATE RESULT PARSE TABLE DESCRIPTOR
57 01 9A 05DF 824 MOVZBL #1,R7 :SET ITEM TYPE TO END OF LINE
58 F486 CA D0 05E2 825 MOVL WRK_L_EXPANDPTR(R10),R8 :SET LENGTH OF ITEM
FA16' 30 05E7 826 BSBW DCL$GENDESCR :SET STARTING ADDRESS OF ITEM
05 05EA 827 CLRBIT PRC_V_IND,PRC_W_FLAGS(R1) :GENERATE RESULT PARSE DESCRIPTOR
05EE 828 RSB :ENABLE INDIRECT FILE INTERPRETATION
05EF 829 :
```



```
05EF 831 .SBTTL GET INTERNAL ROUTINE INDEX
05EF 832
05EF 833 DCL$LOCATE_INTERNAL - GET INDEX FROM INTERNAL ROUTINE TABLE
05EF 834
05EF 835 THIS ROUTINE SEARCHES THE TABLE OF INTERNAL ROUTINES FOR A SPECIFIED STRING.
05EF 836 IF FOUND, THE INDEX OF THAT STRING IS RETURNED.
05EF 837
05EF 838 INPUTS:
05EF 839
05EF 840 R1/R2 = ROUTINE NAME DESCRIPTOR
05EF 841
05EF 842 OUTPUTS:
05EF 843
05EF 844 R0 = STATUS
05EF 845 R3 = INDEX NUMBER
05EF 846 R4 IS DESTROYED
05EF 847
05EF 848
05EF 849 DCL$LOCATE_INTERNAL:
05EF 850 MOVQ R1,-(SP) : SAVE R1/R2
05EF 851 MOVQ (R2),-(SP) : GET ROUTINE NAME
05EF 852 BISB #*X80,(SP) : SET HIGH ORDER BIT TO FORCE ROUTINE MATCH
05EF 853 CMPL R1,#8 : ROUTINE LENGTH GREATER THAN 8?
05EF 854 BLEQ 5$ : BRANCH IF NOT
05EF 855 MOVL #8,R1 : SET LENGTH EQUAL 8
05EF 856 5$: MOVL #ROUTINES_LEN,R4 : SET TABLE LENGTH
05EF 857 MATCHC R1,(SP),R4,ROUTINES : FIND SPECIFIED ROUTINE
05EF 858 BNEQ 20$ : BRANCH IF NO MATCH
05EF 859 BICL #7,R2 : ROUND REMAINING LENGTH UP
05EF 860 ADDL #8,R2
05EF 861 SUBL R2,R4
05EF 862 ASHL #-3,R4,R3 : CALCULATE RELATIVE POSITION OF MATCH
05EF 863 MOVL #1,R0 : DIVIDE BY 8 TO GET INDEX INTO R3
05EF 864 10$: CLRQ (SP)+ : SET SUCCESS
05EF 865 MOVL (SP)+,R1 : RESTORE STACK
05EF 866 RSB : RESTORE R1/R2
05EF 867
05EF 868 20$: MOVL #CLIS_INVROUT,R0 : SET INVALID ROUTINE STATUS
05EF 869 BRB 10$ : RETURN
05EF 870
05EF 871 .END
```

7E 51 7D 05EF 850
7E 62 7D 05F2 851
6E 80 8F 88 05F3 852
08 51 D1 05F9 853
51 03 15 05FC 854
54 000001B0 8F D0 05FE 855
00000007'EF 54 6E 51 D0 0601 856
52 07 CA 0611 857
52 08 C0 0613 858
54 52 C2 0616 859
53 54 FD 8F 78 0619 860
50 01 D0 061C 861
51 BE 7C 0621 862
8E 7D 0624 863
05 0626 864
0629 865
062A 866
50 00038912 8F D0 062A 867
F1 11 0631 868
0633 869
0633 870
0633 871

COMMAND
Symbol table

- PROCESS NEXT COMMAND

N 8

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00
4-SEP-1984 23:39:43 [DCL.SRC]COMMAND.MAR;1

Page 23
(12)

\$INTIMAGES	= 000000B6		
\$NAM1\$	= 435349C4		
\$NAM2\$	= 454E4E4F		
BUILD_IMAGE	000002D0	R	03
CLISK-VERB FORE	*****	X	03
CLIS_INVROOT	= 00038912		
CLIS_IVVALU	= 00038088		
CLIS_NOCOMD	= 000380B0		
CLIS_NOLBLS	= 000381E0		
CLIS-NORMAL	= 00030001		
CMDVERB	00000276	R	03
CMD_V_FOREIGN	= 00000002		
CMD_V_IMMED	= 00000003		
CMD_W_FLAGS	= 00000004		
COMMAND	000001F1	R	03
CTLSAG CLITABLE	*****	X	03
DCL\$ABORT	00000504	RG	03
DCL\$ALLDEACMD	*****	X	03
DCL\$ALLOCATE	*****	X	03
DCL\$ALLOC LABEL	*****	X	03
DCL\$ASSIGN	*****	X	03
DCL\$ATTACH	*****	X	03
DCL\$BACKUPCHAR	*****	X	03
DCL\$BACKUPMOVE	*****	X	03
DCL\$CANCEL	*****	X	03
DCL\$CHARERROR	*****	X	03
DCL\$CHECK_AST	0000054F	RG	03
DCL\$CLOSE	*****	X	03
DCL\$CMDSTART	00000011	RG	03
DCL\$CONNECT	*****	X	03
DCL\$CONTINUE	*****	X	03
DCL\$CRETABLE	*****	X	03
DCL\$DEALGOTO	*****	X	03
DCL\$DEALLOCAT	*****	X	03
DCL\$DEASSIGN	*****	X	03
DCL\$DEBUG	*****	X	03
DCL\$DECK	00000547	RG	03
DCL\$DEFINE	*****	X	03
DCL\$DEFKEY	*****	X	03
DCL\$DELKEY	*****	X	03
DCL\$DELSYM	*****	X	03
DCL\$DEPOSIT	*****	X	03
DCL\$DISABLE	00000557	RG	03
DCL\$DISCONNECT	*****	X	03
DCL\$EOD	0000053A	RG	03
DCL\$EOJ	00000504	RG	03
DCL\$EQUATE	*****	X	03
DCL\$ERRORMSG	*****	X	03
DCL\$EVALUATE	*****	X	03
DCL\$EXAMINE	*****	X	03
DCL\$EXIT	*****	X	03
DCL\$EXTIMAGE	*****	X	03
DCL\$FLUSH	00000580	RG	03
DCL\$FORCEEXIT	*****	X	03
DCL\$GENDESCR	*****	X	03
DCL\$GENEOL	*****	X	03
DCL\$GETCHAR	*****	X	03

DCL\$GETOKEN	*****	X	03
DCL\$GOTO	*****	X	03
DCL\$HYPHEN	*****	X	03
DCL\$IF	*****	X	03
DCL\$INPUT	*****	X	03
DCL\$INQUIRE	*****	X	03
DCL\$LOCATE_INTERNAL	000005EF	R	03
DCL\$LOCKED_STATE	*****	X	03
DCL\$LOGOUT	*****	X	03
DCL\$MARK	*****	X	03
DCL\$MARKEDTOKEN	*****	X	03
DCL\$MCR	*****	X	03
DCL\$MOVCHAR	*****	X	03
DCL\$ON	*****	X	03
DCL\$OPEN	*****	X	03
DCL\$PARSE COMMAND	*****	X	03
DCL\$PROCFILE	*****	X	03
DCL\$READ	*****	X	03
DCL\$RECALL	*****	X	03
DCL\$RESTART	00000000	RG	03
DCL\$RUN	*****	X	03
DCL\$SEARCH VERB	*****	X	03
DCL\$SETCHAR	*****	X	03
DCL\$SETCTLY	*****	X	03
DCL\$SETDEFAULT	*****	X	03
DCL\$SETFLUSH	*****	X	03
DCL\$SETKEY	*****	X	03
DCL\$SETNBLK	*****	X	03
DCL\$SETON	*****	X	03
DCL\$SETPROMPT	*****	X	03
DCL\$SETPROT	*****	X	03
DCL\$SETUIC	*****	X	03
DCL\$SETVERIFY	*****	X	03
DCL\$SET STATUS	*****	X	03
DCL\$SHOWDEF	*****	X	03
DCL\$SHOWKEY	*****	X	03
DCL\$SHOWPROT	*****	X	03
DCL\$SHOWQUOTA	*****	X	03
DCL\$SHOWSTAT	*****	X	03
DCL\$SHOWSYMBL	*****	X	03
DCL\$SHOWTIME	*****	X	03
DCL\$SHOWTRAN	*****	X	03
DCL\$SORT TOKENS	*****	X	03
DCL\$SPAWN	*****	X	03
DCL\$SPECIAL	*****	X	03
DCL\$STOP	*****	X	03
DCL\$SUBASSIGN	*****	X	03
DCL\$SYM STRING	*****	X	03
DCL\$WAIT	*****	X	03
DCL\$WRITE	*****	X	03
DEVSV_RND	= 0000001C		
DEVSV_TRM	= 00000002		
ERROR	00000282	R	03
ERROR_EXIT	00000337	R	03
EXECXT	000002F0	R	03
FORVERB	00000255	R	03
IMAGECASE	00000349	R	03

COMMAND
Symbol table

- PROCESS NEXT COMMAND

B 9

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00
4-SEP-1984 23:39:43 [DCL.SRC]COMMAND.MAR;1

Page 24
(12)

IMG_K_ALLOCATE	= 00000080		
IMG_K_ASSIGN	= 00000081		
IMG_K_ATTACH	= 000000A9		
IMG_K_CANCEL	= 000000AB		
IMG_K_CLOSE	= 00000082		
IMG_K_CONNECT	= 000000B4		
IMG_K_CONTINUE	= 00000083		
IMG_K_CRETABLE	= 000000B3		
IMG_K_DEALLOCAT	= 00000084		
IMG_K_DEASSIGN	= 00000085		
IMG_K_DEBUG	= 00000086		
IMG_K_DECK	= 00000087		
IMG_K_DEFINE	= 00000088		
IMG_K_DEFKEY	= 000000AE		
IMG_K_DELKEY	= 000000B0		
IMG_K_DELSYM	= 0000008A		
IMG_K_DEPOSIT	= 00000089		
IMG_K_DISCONNECT	= 000000B5		
IMG_K_EOD	= 0000008B		
IMG_K_EOJ	= 000000AA		
IMG_K_EXAMINE	= 0000008C		
IMG_K_EXIT	= 0000008D		
IMG_K_EXTIMAGE	= 0000008E		
IMG_K_GOTO	= 0000008F		
IMG_K_IF	= 00000090		
IMG_K_INQUIRE	= 00000091		
IMG_K_LOGOUT	= 00000092		
IMG_K_MCR	= 00000093		
IMG_K_ON	= 00000094		
IMG_K_OPEN	= 00000095		
IMG_K_READ	= 00000096		
IMG_K_RECALL	= 000000AD		
IMG_K_RUN	= 00000097		
IMG_K_SETCTLY	= 00000098		
IMG_K_SETDEFAULT	= 00000099		
IMG_K_SETFLUSH	= 000000B2		
IMG_K_SETKEY	= 000000B1		
IMG_K_SETON	= 0000009A		
IMG_K_SETPROMPT	= 000000AC		
IMG_K_SETPROT	= 0000009B		
IMG_K_SETUIC	= 0000009C		
IMG_K_SETVERIFY	= 0000009D		
IMG_K_SHOWDEF	= 0000009E		
IMG_K_SHOWKEY	= 000000AF		
IMG_K_SHOWPROT	= 0000009F		
IMG_K_SHOWQUOTA	= 000000A0		
IMG_K_SHOWSTAT	= 000000A1		
IMG_K_SHOWSYMBL	= 000000A2		
IMG_K_SHOWTIME	= 000000A3		
IMG_K_SHOWTRAN	= 000000A4		
IMG_K_SPAWN	= 000000AB		
IMG_K_STOP	= 000000A5		
IMG_K_WAIT	= 000000A6		
IMG_K_WRITE	= 000000A7		
IMMED	000002FA	R	03
LOGOUT	00000000	R	02
NOCMD	0000027B	R	03

NULLNODE	000001CE	R	03
PRCVERB	00000285	R	03
PRC_B_CONTINUE	000000F3		
PRC_B_DEFRADIX	000000AE		
PRC_B_EXMDEPMOD	000000AD		
PRC_B_EXMDEPWID	000000AC		
PRC_B_EXONLYL	0000012D		
PRC_B_FLAGS2	000000AF		
PRC_B_IMGFLAG	00000078		
PRC_B_OUTFLAGS	0000012C		
PRC_B_PROMPTLEN	000000F0		
PRC_C_LENGTH	00000534		
PRC_G_COMMANDS	00000133		
PRC_G_PROMPT	000000F4		
PRC_K_LENGTH	00000534		
PRC_L_CURRKEY	00000048		
PRC_L_EXMDEPADR	000000A8		
PRC_L_EXTARG	00000094		
PRC_L_EXTBLK	0000008C		
PRC_L_EXTCOD	0000009C		
PRC_L_EXTHND	00000090		
PRC_L_EXTPRM	00000098		
PRC_L_IDFLNK	000000BC		
PRC_L_IMGACTSTS	00000080		
PRC_L_INDCLOCK	0000007C		
PRC_L_INDEPTH	0000005C		
PRC_L_INDFAB	0000001C		
PRC_L_INDINPRAB	00000014		
PRC_L_INDOUTRAB	00000018		
PRC_L_INPRAB	00000008		
PRC_L_LASTKEY	0000004C		
PRC_L_LSTSTATUS	000000B0		
PRC_L_ONCTLY	000000B8		
PRC_L_ONERROR	0000006C		
PRC_L_OUTOFBAND	000000B4		
PRC_L_OUTRAB	0000000C		
PRC_L_OUTRABCTX	00000118		
PRC_L_PPFLIST	00000070		
PRC_L_RECALLPTR	0000012F		
PRC_L_RESTART	00000058		
PRC_L_SAVAP	00000000		
PRC_L_SAVFP	00000004		
PRC_L_SEVERITY	00000050		
PRC_L_SPWN	000000C0		
PRC_L_STACKLM	000000A4		
PRC_L_STACKPT	000000A0		
PRC_L_STATUS	00000054		
PRC_L_STS	00000084		
PRC_L_STV	00000088		
PRC_L_SYMBOL	00000060		
PRC_L_TMBX	00000074		
PRC_L_TRLIST	00000010		
PRC_M_DISABL	= 00000004		
PRC_M_FLUSH	= 00000040		
PRC_M_IND	= 00000020		
PRC_M_RUNDEF	= 00000004		
PRC_Q_ALLOCREG	00000020		

COMMAND
Symbol table

- PROCESS NEXT COMMAND

C 9

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00
4-SEP-1984 23:39:43 [DCL.SRC]COMMAND.MAR;1

Page 25
(12)

```

PRC_Q_COMMAND      000000E0
PRC_Q_FLUSHTIME    000000D0
PRC_Q_GLOBAL       00000028
PRC_Q_IMAGENAME    000000D8
PRC_Q_KEYPAD       00000040
PRC_Q_LABEL        00000030
PRC_Q_LOCAL        00000038
PRC_Q_SAVEPRIV     000000E8
PRC_T_OUTDVI       0000011C
PRC_V_CHAIN        = 00000001
PRC_V_CMD          = 00000000
PRC_V_CNTRL        = 00000001
PRC_V_DISABL       = 00000002
PRC_V_FLUSH        = 00000006
PRC_V_GOTO         = 00000004
PRC_V_HANGUP       = 0000000C
PRC_V_IND          = 00000005
PRC_V_MODE         = 00000006
PRC_V_PRIV         = 00000004
PRC_V_RUNDEF       = 00000002
PRC_V_YLEVEL       = 0000000B
PRC_W_ASTIOSB      000000C6
PRC_W_ASTRETN      000000C8
PRC_W_ASTSTATUS    000000C4
PRC_W_ATTMBX       0000007A
PRC_W_FLAGS        00000068
PRC_W_INPCHAN      00000064
PRC_W_ONLEVEL      0000006A
PRC_W_OUTIFI       00000114
PRC_W_OUTISI       00000116
PRC_W_OUTMBXCHN    000000CA
PRC_W_OUTMBXREF    000000CE
PRC_W_OUTMBXSIZ    000000CC
PRC_W_PMPCTCTRL    000000F1
PRC_W_WAITIOSB     00000066
PROCFORM          = 000005AF R 03
PSLSV_CURMOD      = 00000018
PTR_B_LEVEL        00000004
PTR_B_NUMBER       00000005
PTR_B_PARMCNT      00000006
PTR_B_VALUE        00000000
PTR_C_LENGTH       0000000C
PTR_K_ENDLINE      = 00000004
PTR_K_LENGTH       = 0000000C
PTR_K_PARAMETER    = 00000003
PTR_L_DESCR        00000000
PTR_L_ENTITY       00000008
RABSL_CTX         = 00000018
ROUTINES           = 00000007 R 02
ROUTINES_LEN      = 000001B0
SS$_ILLSER        ***** X 03
SYMBOL            000001DB R 03
SYM_B_FLAGS        0000000B
SYM_B_NONUNIQUE    0000000B
SYM_B_TYPE         0000000A
SYM_L_BL          00000004
SYM_L_FL          00000000

```

```

SYM_T_SYMBOL      0000000C
SYM_W_SIZE        00000008
WRK_B_CMDOPT      FFFFFFFC3
WRK_B_MAXPARM     FFFFFFFD0
WRK_B_MINPARM     FFFFFFFD1
WRK_B_PARMCNT     FFFFFFFCE
WRK_B_PARMSUM     FFFFFFFCF
WRK_B_RECALLCNT   FFFFFFFC5
WRK_B_VALLEV      FFFFFFFC4
WRK_B_VERBTYP     FFFFFFFC2
WRK_C_LENGTH      FFFFF486
WRK_C_SCRSTKSIZ   = 00000200
WRK_G_BUFFER      FFFFFFF492
WRK_G_INPBUF      FFFFFFF896
WRK_G_RESULT      FFFFFFF9B6
WRK_K_LENGTH      FFFFF486
WRK_L_CHARPTR     FFFFF48E
WRK_L_DISALLOW    FFFFFFFE6
WRK_L_ERRORRTN    FFFFFFF9AE
WRK_L_EXPANDPTR   FFFFFFF486
WRK_L_IMAGE       FFFFFFFE2
WRK_L_MARKPTR     FFFFF48A
WRK_L_PAROUT      FFFFFFFD2
WRK_L_PMPADDR     FFFFFFF9A2
WRK_L_PROMPTRTN   FFFFFFF9A6
WRK_L_PROPTR      FFFFFFFC6
WRK_L_QUABLK      FFFFFFFCA
WRK_L_READRTN     FFFFFFF9AA
WRK_L_RECALLPTR   FFFFFFFEA
WRK_L_RSLND       FFFFFFFB6
WRK_L_RSLNXT      FFFFFFFBA
WRK_L_SAVAP       FFFFFFFF8
WRK_L_SAVFP       FFFFFFFFC
WRK_L_SAVSP       FFFFFFFF4
WRK_L_SIGNALRTN   FFFFFFFD6
WRK_L_SPECRTN     FFFFFFF9B2
WRK_L_TAB_VEC     FFFFFFFDE
WRK_L_VERB        FFFFFFFBE
WRK_V_CLIRtn      = 00000000
WRK_V_COMMAND     = 00000001
WRK_W_FLAGS       FFFFFFFF0
WRK_W_FLAGS2      FFFFFFFF2
WRK_W_IMGCHAN     FFFFFFFEE
WRK_W_PMPLEN      FFFFF99E
XXX$ALLOCATE      000003C0 R 03
XXX$ASSIGN        000003C6 R 03
XXX$ATTACH        000004B6 R 03
XXX$CANCEL        000004C2 R 03
XXX$CLOSE         000003CC R 03
XXX$CONNECT       000004F8 R 03
XXX$CONTINUE      000003D2 R 03
XXX$CRETABLE      000004F2 R 03
XXX$DEALLOCAT     000003D8 R 03
XXX$DEASSIGN      000003DE R 03
XXX$DEBUG         000003E4 R 03
XXX$DECK          000003EA R 03
XXX$DEFINE        000003F0 R 03

```


COMMAND
Symbol table

- PROCESS NEXT COMMAND

D 9

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00
4-SEP-1984 23:39:43 [DCL.SRC]COMMAND.MAR;1

Page 26
(12)

XXX\$DEFKEY	000004D4	R	03
XXX\$DELKEY	000004E0	R	03
XXX\$DELSYM	000003FC	R	03
XXX\$DEPOSIT	000003F6	R	03
XXX\$DISCONNECT	000004FE	R	03
XXX\$EOD	00000402	R	03
XXX\$EOJ	000004BC	R	03
XXX\$EXAMINE	00000408	R	03
XXX\$EXIT	0000040E	R	03
XXX\$EXTIMAGE	00000414	R	03
XXX\$GOTO	0000041A	R	03
XXX\$IF	00000420	R	03
XXX\$INQUIRE	00000426	R	03
XXX\$LOGOUT	0000042C	R	03
XXX\$MCR	00000432	R	03
XXX\$ON	00000438	R	03
XXX\$OPEN	0000043E	R	03
XXX\$READ	00000444	R	03
XXX\$RECALL	000004CE	R	03
XXX\$RUN	0000044A	R	03
XXX\$SETCTLY	00000450	R	03
XXX\$SETDEFAULT	00000456	R	03
XXX\$SETFLUSH	000004EC	R	03
XXX\$SETKEY	000004E6	R	03
XXX\$SETON	0000045C	R	03
XXX\$SETPROMPT	000004C8	R	03
XXX\$SETPROT	00000462	R	03
XXX\$SETUIC	00000468	R	03
XXX\$SETVERIFY	0000046E	R	03
XXX\$SHOWDEF	00000474	R	03
XXX\$SHOWKEY	000004DA	R	03
XXX\$SHOWPROT	0000047A	R	03
XXX\$SHOWQUOTA	00000480	R	03
XXX\$SHOWSTAT	00000486	R	03
XXX\$SHOWSYMBL	0000048C	R	03
XXX\$SHOWTIME	00000492	R	03
XXX\$SHOWTRAN	00000498	R	03
XXX\$SPAWN	000004B0	R	03
XXX\$STOP	0000049E	R	03
XXX\$WAIT	000004A4	R	03
XXX\$WRITE	000004AA	R	03
\$\$	= 000000EF		

! Psect synopsis !

PSECT name

Allocation

PSECT No.

Attributes

. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
\$AB\$\$	FFFFFFFFC (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
DCL\$ZDATA	000001B7 (439.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE
DCL\$ZCODE	00000633 (1587.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	18	00:00:00.04	00:00:01.49
Command processing	97	00:00:00.64	00:00:06.46
Pass 1	361	00:00:13.86	00:00:43.10
Symbol table sort	0	00:00:01.58	00:00:04.96
Pass 2	181	00:00:03.27	00:00:09.98
Symbol table output	47	00:00:00.32	00:00:01.20
Psect synopsis output	4	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	708	00:00:19.75	00:01:07.23

The working set limit was 1200 pages.
87010 bytes (170 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1080 non-local and 49 local symbols.
871 source lines were read in Pass 1, producing 26 object records in Pass 2.
43 pages of virtual memory were used to define 27 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	13
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	19

1101 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:COMMAND/OBJ=OBJ\$:COMMAND MSRC\$:COMMAND/UPDATE=(ENH\$:COMMAND)+EXECML\$/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSBLDMLB/LIB

0069 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INTDEF
SOL

COMMAND
LIS

DESCRVAL
LIS

CLIMAC
MAR

CLIMSG
LIS

CONNECT
LIS

DCLPARSE
LIS

EXAMDEP
LIS

EXIT
LIS

INTIMAGES
MAR

DCXSTART
LIS

CLIGBL
LIS

DISALLOW
LIS

CLINT
LIS

CANCEL
LIS